🥳 WasmEdge (formerly SSVM) is now a CNCF Project

• 5 minutes to read

The WasmEdge Runtime (formerly SSVM) is a lightweight and high-performance WebAssembly (WASM) VM optimized for edge computing. It is widely used in scenarios ranging from cloud serverless functions, SaaS, blockchain smart contracts, IoT, to automotive real time applications.


The TOC met today to review the applications for projects wishing to be included as sandbox projects. The next scheduled meeting for sandbox inclusion review is June 22nd, 2021.

Vineyard - passes with a majority vote of the TOC
WasmEdge Runtime - passes with a majority vote of the TOC
ChaosBlade - passes with a majority vote of the TOC
YARP - Yet Another Reverse Proxy - Presentation to SIG Network to review, is this cloud native and how is this different from existing CNCF projects?
Kubeinvaders - not included
KubePhaa - not included
Service Mesh Performance - Request from TOC to either combine or align these projects with Service Mesh Interface
Meshery - Request from TOC to either combine or align these projects with Service Mesh Interface
Fluid - passes with a majority vote of the TOC
Submariner - passes with a majority vote of the TOC
Pixie - Presentation requested for SIG Observability
Antrea - passes with a majority vote of the TOC

WasmEdge has recently been accepted by the CNCF Foundation as a sandbox project, which is the world's first official CNCF WebAssembly runtime project. We envision that it will fill an important role in CNCF's open source cloud computing landscape as a mission critical lightweight runtime for edge clouds.

> Source code: https://github.com/WasmEdge/WasmEdge

The beginning

Twenty five years ago, Java was first invented as a programming language for browser widgets. But it eventually found success on the server side. Later, JavaScript repeated the same migration path from browser to server / cloud via Node.js. Today, WebAssembly is again in the middle of a migration from browser to the cloud. WasmEdge is on the forefront of this historic migration.

History rhythms but never repeats. While the WebAssembly migration from browser to cloud is driven by some of the same factors as Java and JavaScript before it, such as adoption by young developers, and a popular programming language (Rust), WebAssembly is also uniquely well suited to handle the modern cloud computing workload. Specially, WebAssembly's lightweight design and advanced security model, together with Rust language's memory safety, enables it to be used in high performance and mission critical applications in the cloud. Especially on the edge cloud, IoT devices, and automobiles, where traditional containers and VMs are too heavy and slow, WebAssembly could potentially become the dominant runtime technology.

WasmEdge, formerly SSVM, is an open source WebAssembly VM optimized for the modern cloud and edge devices. According to a research paper published on IEEE Software, with advanced AOT compiler support, WasmEdge is the fastest runtime on the market today.

Key Features of WasmEdge

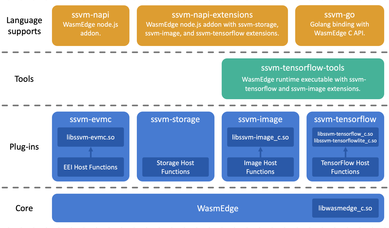
WasmEdge is fully compatible with the W3C WebAssembly standard. Out of the box, it is supported by standard language and compiler tool chains, such as LLVM, Rustc and emscripten. WasmEdge is differentiated for its support for extensions, especially extensions for edge computing.

For starters, WasmEdge supports W3C optional WebAssembly features and proposals, such as WebAssembly System Interface(WASI) spec, Reference type, Bulk memory operations, and SIMD. We are also exploring the wasi-socket proposal to support network access in WebAssembly programs.

Furthermore, WasmEdge supports non-standard extensions designed for specific application scenarios.

- Tensorflow. Developers can write Tensorflow inference functions using a simple Rust API, and then run the function securely and at native speed inside WasmEdge. We are working on supporting other AI frameworks.
- Storage. The WasmEdge storage interface allows WebAssembly programs to read and write a key-value store.
- Command interface. WasmEdge enables webassembly functions to execute native commands in the host operating system. It supports passing arguments, environment variables, STDIN / STDOUT pipes, and security policies for host access.
- Ethereum. The WasmEdge Ewasm extension supports Ethereum smart contracts compiled to WebAssembly. It is a leading implementation for Ethereum flavored WebAssembly (Ewasm).
- Substrate. The Pallet allows WasmEdge to act as an Ethereum smart contract execution engine on any Substrate based blockchains.

Last but not least, WasmEdge is a "cloud-native" WebAssembly VM. It supports the OCI (Open Container Initiative)specification, which will allow WasmEdge instances to be managed by cloud-native orchestration tools such as Kubernetes.



WasmEdge is formerly called SSVM.

Use Cases

WasmEdge is optimized for edge and embedded use cases. It turns your software or hardware product into a developer platform. Here are some specific use cases.

- A Jamstack application consists of a static frontend with JavaScript to interact with back-end APIs. It is a trendy modern web application architecture. The frontend static files can be distributed over CDNs, and the backend functions can be hosted on edge nodes. The cloud-based WasmEdge hosts secure and high-performance backend serverless functions for Jamstack apps, especially on the Edge cloud.
  - Example: add a watermark to any image on your web app.
  - Example: serverless Tensorflow functions for Tencent Cloud.
- SaaS applications often need to be tailored or customized "on the edge" for customer requirements. With WasmEdge, SaaS applications can directly embed and execute user-submitted code as part of the workflow (eg, as a callback function to handle events from the SaaS app).
  - Example: the Lark / Feishu application platform could embed user-submitted serverless functions via WasmEdge to respond to messages (ie conversation bot).
  - Example: WasmEdge runs custom code to process events in IoT streaming data framework YoMo.
- WasmEdge is adapted to run on a variety of embedded and real-time operating systems for edge devices. That allows developers to write high-performance applications once, in Rust or C, and run them safely on many edge device platforms.
  - Example: RISC-V stack from RIOS Lab.
  - Ongoing: Porting WasmEdge to the SeL4 real-time OS
  - Upcoming; WasmEdge could be used as an RTOS code runtime for software modules in autonomous cars.
- Blockchain smart contracts are user-submitted code executed by all nodes in the network. WasmEdge is a smart contract execution engine on leading blockchain projects. Example: Ethereum flavored WASM smart contracts on Substrate and Polkadot.

The Future

The WebAssembly ecosystem is still in its early days. Hosted by CNCF, WasmEdge aims to become an open source "reference implementation" of WebAssembly and its edge-related extensions. The community will be able to experiment with new extension ideas on WasmEdge first, and then standardize the successful extensions. The WasmEdge community will also contribute to compiler toolchains and SDKs to support WebAssembly and WasmEdge extensions in more programming languages.

Join us in the WebAssembly revolution!

👉 Slack Channel: #wasmedge on CNCF slack channel

👉 Mailing list: Send an email to WasmEdge@googlegroups.com

👉 Twitter: @realwasmedge

👉 Be a contributor: checkout our wish list to start contributing!

Rust Go WebAssembly #WasmEdge #open-source #CNCF #Cloud Native

SECOND STATE

A high-performance, extensible, and hardware optimized WebAssembly Virtual Machine for automotive, cloud, AI, and blockchain applications

second-state @secondstateinc